1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский политехнический университет Петра Великого
3. —
4. Институт компьютерных наук и технологий
5. **Кафедра «Информационная безопасность компьютерных систем»**

**ОТЧЕТ**

**по курсовой работе**

1. по дисциплине «Вычислительная математика»
2. Выполнил
3. студент гр. 23508/4 Е.Г. Проценко
4. Проверил
5. профессор С.М. Устинов
6. Санкт-Петербург
7. 2016

# Формулировка задания (вариант 13)

Для решения нелинейной краевой задачи относительно на интервале

,

,

может быть использован следующий подход.

Исходное уравнение переписываем в виде

Отсюда, (1),

где – некоторая константа. Поскольку , то .

Если бы мы могли вычислить , то исходная задача свелась бы к задаче Коши, легко решаемой с помощью подпрограммы *RKF45*.

Интегрирование уравнения (1) дает .

Используя граничное условие , получим уравнения для : , которое может быть решено с помощью подпрограммы QUANC8 и ZEROIN.

Реализовать этот подход к решению задачи. Оценить погрешность результатов и погрешность определяемую неточностью в исходных данных.

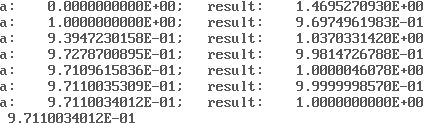
# Результаты работы

## Нахождение параметра

Единственное, что нам мешает решать сразу дифференциальное уравнение, используя подпрограмму , это то, что параметр нам неизвестен. Для его нахождения предложено использовать подпрограмму . Т.к. в крайнюю названную программу поступает функция содержащая интеграл, то мы будем использовать внутри той функции, которая отправляется в .

|  |
| --- |
| а := zeroin(ax,bx,tol,@F1);  ----------------------------------  function F1(new\_a: float) : float;  begin  a := new\_a;  abserr := 0;  relerr := 0.00001;  quanc8(@F2, 0, 1, abserr, relerr, result, errest, nofun, flag);  writeln('a: ', a, '; result: ', result);  F1 := result - 1;  end;  -----------------------------------  function F2(y: float) : float;  begin  F2 := (1/(sqrt(2) \* power((a + power(y, 3) / 3 - y), 1/2)));  end; |

Результат работы :



0.97110034012 – это , полученное от подпрограммы .

## Решение дифференциального уравнения второго порядка

Получив , мы можем подставить его в соответствующую формулу. Помимо этого нам стал известен .

Следующим шагов является решение дифференциального уравнения второго порядка. У нас есть подпрограмма , которая работает с дифференциальными уравнениями, но первого порядка. Поэтому на нужно используя замену разложить дифференциальное уравнение первого порядка на 2 дифференциального уравнения первого порядка:

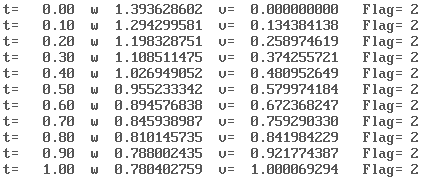
По коду: .

|  |
| --- |
| procedure F4(t: float; var z, dz: floatvector);  begin  dz[1] := power(z[2], 2) - 1;  dz[2] := z[1];  end; |

Начальные условия:

Теперь можно воспользоваться подпрограммой .

Результат работы :



## Влияние погрешности исходных данных на решение

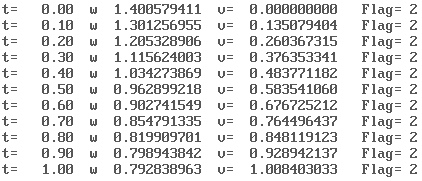
Касательно погрешности подпрограмм и можно сказать, что вычисления получены добротные, , . Трогать их не будем.

Но что будет, если мы получим параметр с погрешностью .

|  |
| --- |
| a := zeroin(ax,bx,tol,@F1);  a := a \* 1.01; |

Новое .

Результат работы RKF45:

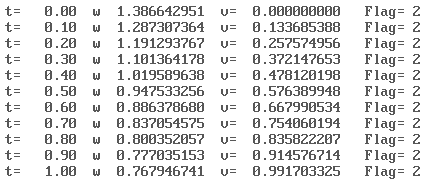


Теперь посмотрим, что будет, если параметр меньше того, что мы получили ранее на .

|  |
| --- |
| a := zeroin(ax,bx,tol,@F1);  a := a \* 0.99; |

Новое .

Результат работы RKF45:



# Вывод

Зная, что , получаем, что при том , что мы получили изначально, можно сделать вывод, что погрешность появилась только в 5-ом разряде.

При увеличении и уменьшении на , погрешность перепрыгнула с пятого порядка на третий. Можно с уверенностью сказать, что такая система не является очень то устойчивой, т.к. при изменении исходных данных всего на 1% мы получили погрешность в 100 раз больше, чем раньше.

# Приложение

Листинг написанной программы:

uses FMM, CRT, MATH;

label rinse;

Var

a, a\_etalon: float;

ax, bx, tol: float;

y: float;

repits:integer;

z, zp: floatvector;

t, tout, tfinal, tprint: float;

iflag: integer;

work: rvecn;

iwork: ivec5;

abserr, relerr: float;

nofun: longint;

flag: float;

result: float;

errest: float;

{$F+}

function F2(y: float) : float;

begin

F2 := (1/(sqrt(2) \* power((a + power(y, 3) / 3 - y), 1/2)));

end;

function F1(new\_a: float) : float;

{Var}

begin

a := new\_a;

abserr := 0;

relerr := 0.0000001;

quanc8(@F2, 0, 1, abserr, relerr, result, errest, nofun, flag);

writeln('a: ', a, '; result: ', result);

F1 := result - 1;

end;

procedure F4(t: float; var z, dz: floatvector);

begin

dz[1] := power(z[2], 2) - 1;

dz[2] := z[1];

end;

{$F-}

begin

clrscr;

{ZEROIN + QUANC8}

ax := 0;

bx := 1;

tol := 1e-10;

a := zeroin(ax,bx,tol,@F1);

writeln(a);

readln;

repit:

{RFK45}

t := 0;

tfinal := 1;

tout := t;

tprint := 0.1;

relerr := 0.000001;

abserr := 0;

iflag := 1;

z[1] := sqrt(2 \* a);

z[2] := 0;

rinse:

rkf45(@F4,2,z,t,tout,relerr,abserr,iflag,work,iwork);

writeln(' t= ',t:6:2, ' w',z[1]:13:9,' v=',

z[2]:13:9,' Flag=',iflag:2);

case iflag of

1, 8 : exit;

2 : begin

tout := t + tprint;

if t < tfinal then goto rinse

end;

4 : goto rinse;

5 : begin

abserr := 1E-9;

goto rinse

end;

6 : begin

relerr := 10 \* relerr;

iflag := 2;

goto rinse

end;

7 : begin

iflag := 2;

goto rinse

end;

end;

readln;

end.